The Stata Journal

H. Joseph Newton
Department of Statistics
Texas A & M University
College Station, Texas 77843
979-845-3142
979-845-3144 FAX
jnewton@stata-journal.com

Nicholas J. Cox
Department of Geography
University of Durham
South Road
Durham City DH1 3LE
United Kingdom
n.j.cox@stata-journal.com

The *Stata Technical Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press, and Stata is a registered trademark of Stata Corporation.

The Stata Journal (2003)
3, Number 1, pp. 71–80

From the help desk: Bootstrapped standard errors

Weihua Guan
Stata Corporation

Abstract. Bootstrapping is a nonparametric approach for evaluating the distribution of a statistic based on random resampling. This article illustrates the bootstrap as an alternative method for estimating the standard errors when the theoretical calculation is complicated or not available in the current software.

Keywords: st0034, bootstrap, cluster, nl, instrumental variables

1 Introduction

Suppose that we have a random sample from an unknown (possibly multivariate) distribution F, and we want to make statistical inferences about a parameter θ. The traditional parametric approach depends upon strong distributional assumptions of F. Given the form of F, analytical formulas can be derived for an estimator, $\widehat{\theta}$, and hence its standard error. While a consistent estimator may be easy to obtain, the formula for the standard error is sometimes more difficult, or possibly even mathematically intractable. Moreover, the sampling distribution of $\widehat{\theta}$ may not be of any known standard distribution.

Bootstrapping is a nonparametric approach that permits one to avoid the theoretical calculation. It relies upon the assumption that the current sample is representative of the population, and therefore, the empirical distribution function $\widehat{F}$ is a nonparametric estimate of the population distribution F. From the sample dataset, the desired statistic, $\widehat{\theta}$, can be calculated as an empirical estimate of the true parameter. To measure the precision of the estimates, a bootstrapped standard error can be calculated in the following way:

1. draw random samples with replacement repeatedly from the sample dataset;
2. estimate the desired statistic corresponding to these bootstrap samples, which forms the sampling distribution of $\widehat{\theta}$; and
3. calculate the sample standard deviation of the sampling distribution.

This approach utilizes the same theory underlying Monte Carlo simulation methods, except it utilizes resamples from the original data rather than from the population. When the sample size is large, the bootstrapping estimates will converge to the true parameters as the number of repetitions increases.

In this paper, several applications of bootstrapping procedures are presented to obtain standard errors. Although the real utility of the bootstrap method is in cases where a model-based solution is not currently available, for all of the examples presented here, at least one model-based solution exists. This was done so that the bootstrap solution could be compared with a known solution. To evaluate the results, we use Monte Carlo simulation, and the procedure is as follows:

1. define the population and draw random samples of size n;
2. use `bootstrap` to obtain bootstrapping estimates;
3. compute the estimates using a model-based method;
4. repeat step 1–3; and
5. calculate the proportions of Type I error in these methods.

Steps 1–4 are implemented using the `simulate` command.

2 Robust standard errors with clustered data

The assumption that error terms are independently and identically distributed (i.i.d.) is often critical in statistical analysis. However, this assumption may not always hold, and a statistical method will fail to give satisfactory results. Suppose, for example, we perform an experiment where we randomly select a group of individuals and then repeatedly measure each individual's blood pressure at uniform intervals of time over a period of several days. Though the individuals are randomly selected and can therefore be assumed to be independent of each other, the blood pressures of the same individual may be correlated. Having this prior knowledge regarding the sampling design, we know we must employ statistical methods that are able to account for the within-individual correlation.

2.1 Simple linear models

Let us first consider a simple linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

The OLS estimate of the coefficient is $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, and the estimate of variance is $s^2(\mathbf{X}'\mathbf{X})^{-1}$, where s^2 is the mean square error, when the samples are i.i.d. For clustered data, the OLS estimator of the coefficients is still consistent, but the conventional estimates of variances yield incorrect coverage probability. Stata has implemented a robust estimator (also called the Huber/White/sandwich estimator), obtained by specifying the `cluster()` or `robust` option to `regress`. In the presence of heteroskedasticity, i.e., the data are independent but not identically distributed, `robust` can be specified

to give standard errors that are valid for statistical inference. Specifying `cluster` will further relax the assumption of independence within clusters. Alternatively, we may apply bootstrapping techniques to obtain estimates for the variances.

The following Monte Carlo simulation draws random samples from a linear model, where there exists two explanatory variables `x1` and `x2`. The random noise, `e`, is drawn as normally distributed. `z` stands for the individual effects, varying across clusters, such that the assumption of independence will not hold at the observation level. All of the coefficients, including the constant term, are set to one in the simulation. We obtain the estimates through OLS regression and compare the coverage of robust standard errors with that of bootstrapped standard errors.

The simulation program is as follows:

```
program regclus, rclass                       // to be called by -simulate-
        version 8
        drop _all
        set obs 1000                          // generate 1000 panels
        generate z = invnorm(uniform())       // panel-level baseline
        generate id =_n                       // panel identification variable
        expand 5                              // generate 5 observations per panel
        generate x1 = invnorm(uniform())*1.5 // x1 and x2 are explanatory variables
        generate x2 = invnorm(uniform())*1.8
        generate e = invnorm(uniform())       // random noise
        generate y = z + 1 + x1 + x2 + e
                                              // robust standard errors
        regress y x1 x2, cluster(id)
        return scalar x1 = _b[x1]
        return scalar x2 = _b[x2]
        return scalar cons = _b[_cons]
        return scalar sdx1 = _se[x1]
        return scalar sdx2 = _se[x2]
        return scalar sdcons = _se[_cons]
                                              // bootstrapped standard errors
        bootstrap "regress y x1 x2" "_b", reps(1000) cluster(id)
        return scalar bs_sdx1 = _se[b_x1]
        return scalar bs_sdx2 = _se[b_x2]
        return scalar bs_sdcons = _se[b_cons]
end
```

In the simulation, 1,000 clusters are randomly generated, each cluster containing 5 observations. Since the observations are not independent within clusters, the bootstrap samples are drawn in the unit of clusters, defined by `id`. The Monte Carlo simulation is repeated 1,000 times, each having 1,000 bootstrap samples. The Stata command that performs the simulation is

```
simulate "regclus" x1=r(x1) x2=r(x2) cons=r(cons) sdx1=r(sdx1) sdx2=r(sdx2) /*
        */ sdcons=r(sdcons) bs_sdx1=r(bs_sdx1) bs_sdx2=r(bs_sdx2)         /*
        */ bs_sdcons=r(bs_sdcons), reps(1000)
```

The results are summarized below:

```
. summarize, separator(0)
```

Variable	Obs	Mean	Std. Dev.	Min	Max
x1	1000	.9999335	.0134354	.9562576	1.054577
x2	1000	.9995932	.0116971	.9636911	1.03762
cons	1000	1.002357	.0355351	.8854129	1.117827
sdx1	1000	.0133282	.0004434	.0116574	.0147066
sdx2	1000	.0111111	.0003669	.0100201	.0123095
sdcons	1000	.0346499	.0007497	.0321183	.0375121
bs_sdx1	1000	.0133209	.0005364	.0114567	.0151653
bs_sdx2	1000	.0110951	.0004436	.0095247	.0124349
bs_sdcons	1000	.0345745	.0010691	.0309909	.0378581

While the bootstrapped standard errors and the robust standard errors are similar, the bootstrapped standard errors tend to be slightly smaller. Based on the estimated coefficients and standard errors, Wald tests are constructed to test the null hypothesis: $H_0 : \beta = 1$ with a significance level $\alpha = 0.05$. The empirical coverage probability is defined as the fraction of times that the null hypothesis is not rejected. The binomial confidence intervals of the coverage probabilities are calculated using the `ci` command, where the the number of successes is the number of times that the null hypothesis is not rejected and the binomial denominator is 1,000, the number of simulation repetitions.

Table 1: Monte Carlo simulation results for clustered data

		empirical coverage		
variable	$1-\alpha$	conventional	robust	bootstrap
$x1$	0.95	0.955	0.956	0.936
		(0.940, 0.967)*	(0.941, 0.968)	(0.919, 0.950)
$x2$	0.95	0.957	0.953	0.943
		(0.935, 0.963)	(0.938, 0.965)	(0.927, 0.957)
_cons	0.95	0.733	0.939	0.942
		(0.704, 0.760)	(0.922, 0.953)	(0.926, 0.956)

*: binomial exact 95% confidence interval

The first column under empirical coverage gives the coverage probabilities of conventional estimates without using the robust estimation (obtained from another simulation), which shows that the empirical coverage level of the estimated constant term is only 73.3%. The coverage probabilities and the binomial confidence intervals support the conclusion that the bootstrap and robust methods both produce valid estimates of variance such that inference for a specified significance level can be achieved with correct coverage probability.

2.2 Nonlinear least squares regression

In the first example, although the two methods show similar power for the hypothesis tests, the robust estimator is more convenient and requires much less computation. Now, we give an example for which there is no quick Stata solution.

The `nl` command fits a nonlinear model using least squares. However, unlike many other commands, `nl` does not provides a `cluster()` option to handle clustered data. The proposed solution is to make an assumption regarding the distribution of the disturbance in the model. Having done so, one can then employ the method of maximum likelihood to estimate the parameters of the model utilizing Stata's `ml` command. This will allow us to obtain robust variance estimates. As illustrated in the previous section, we may also use bootstrapping to obtain the standard errors.

Suppose that we have a nonlinear model,

$$\mathbf{y} = \beta_0 \left(1 - e^{-\mathbf{x}\beta_1}\right) + \boldsymbol{\epsilon}$$

which can be transformed as

$$\boldsymbol{\epsilon} = \mathbf{y} - \beta_0 \left(1 - e^{-\mathbf{x}\beta_1}\right)$$

If we assume that the error term is normally distributed with mean 0 and variance σ^2, the log-likelihood function can be written as

$$\ln L = \sum_{i=1}^{N} \left(-\frac{1}{2}\ln(2\pi) - \ln\sigma - \frac{1}{2}\frac{\epsilon^2}{\sigma^2} \right)$$

Besides β_0 and β_1, the MLE method estimates an additional parameter σ, the standard deviation of the random errors. Although we assume that the random errors have a normal distribution, this method may still yield consistent estimates for β_0 and β_1. Here we compare the robust estimates of the variances by using `ml` and specifying the `cluster()` option with those obtained from the nonlinear regression and bootstrap method (the bootstrap samples are drawn by clusters).

The Stata programs for `ml` and `nl` are given below.

```
program nlnexpgr
        version 8
        if "`1'" == "?" {                  //    if query call ...
                global S_1 "B0 B1"         //    declare parameters
                global B0=2                //    and initialize them
                global B1=2
                exit
        }
        replace `1'=$B0*(1-exp(-$B1*x))    // otherwise, calculate function
end
```

```
program mlnexpgr
        version 8
        args lnf B1 B0 lnsigma            // the third equation is for
                                             parameter ln(sigma)
        tempvar sigma res
        quietly generate double ‘sigma’ = exp(‘lnsigma’)
        quietly generate double ‘res’ = $ML_y1 - ‘B0’*(1-exp(-‘B1’))
        quietly replace ‘lnf’ = -0.5*ln(2*_pi)-ln(‘sigma’)-0.5*‘res’^2/‘sigma’^2
end
```

In the Monte Carlo simulation, the parameters β_0 and β_1 are both assigned a value of 2. We performed simulations with sample sizes of 100, 500, and 1,000 clusters with 5 observations per cluster. 500 bootstrap samples were drawn for each simulation. The empirical coverage probabilities are calculated from Wald tests and are listed in Table 2. The simulation program is similar to the one in the first example, replacing the linear equation with a nonlinear equation given above. In Table 2, we present the results from

Table 2: Monte Carlo simulation results for clustered data

			empirical coverage		
	parameter	$1-\alpha$	$n^a = 100$	$n = 500$	$n = 1000$
ML	β_0	0.95	0.892	0.919	0.925
			(0.871, 0.911)[b]	(0.900, 0.935)	(0.907, 0.941)
	β_1	0.95	0.881	0.912	0.920
			(0.859, 0.900)	(0.893, 0.929)	(0.901, 0.934)
bootstrap	β_0	0.95	0.987	0.981	0.971
			(0.978, 0.993)	(0.970, 0.989)	(0.959, 0.980)
	β_1	0.95	0.996	0.984	0.977
			(0.990, 0.999)	(0.974, 0.991)	(0.966, 0.985)

[a]: n is the number of clusters in simulation. There are 5 observations per cluster.
[b]: binomial exact 95% confidence interval

Both methods yield unbiased estimates for the two parameters in all experiments. The maximum likelihood estimator consistently yields coverage probabilities that are small, and the bootstrap approach yields coverage probabilities that are greater than 0.95. The maximum likelihood estimators have been proven to be asymptotically efficient, which require large sample size to provide accurate coverage. The bootstrap approach also relies upon large samples such that the samples can simulate the population. Though the binomial confidence intervals do not include the value 0.95 in the three experiments, the results show the tendency that the empirical coverage levels may converge to the value of 0.95 when the number of simulated samples increases. Given the fact that both MLE and bootstrap give asymptotic results, the sample size may play an important role in obtaining appropriate coverage. In addition, the coverage of the bootstrap method is calculated based on the normal approximation of the distribution of the parameters. There are other methods, such as the percentile or bias-corrected method, which may give more appropriate confidence intervals from the bootstrap samples.

3 Two-stage regression with instrumental variables

In a classic linear regression model,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

we assume that the covariates $\mathbf{x}_i, \ldots, \mathbf{x}_k$ are independent of the disturbance term $\boldsymbol{\epsilon}$. In practice, the covariates are sometimes correlated with $\boldsymbol{\epsilon}$. Econometricians refer to such variables as being endogenous. The OLS estimator is not consistent in the presence of endogenous variables. The method of instrumental variables yields a consistent, although biased, estimator. Now, let us extend the model to a general form,

$$\mathbf{y} = f(\mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ may be a categorical or limited dependent variable, and $f()$ is a function of the linear combination $\mathbf{X}\boldsymbol{\beta}$. Here we present an example for the tobit model with endogenous explanatory variables, which is given by

$$\mathbf{y}^* = \mathbf{X}_1\boldsymbol{\beta} + \mathbf{z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

where $\mathbf{X}_1$ contains a set of exogenous covariates and $\mathbf{z}$ is an endogenous covariate. $\mathbf{y}^*$ is the latent dependent variable, and the outcome is only observed when $\mathbf{y}^* > 0$. The disturbance $\boldsymbol{\epsilon}$ is assumed to be normally distributed. $\mathbf{z}$ can be related to a set of instrumental variables

$$\mathbf{z} = \mathbf{X}_1\boldsymbol{\delta}_1 + \mathbf{X}_2\boldsymbol{\delta}_2 + \mathbf{v}$$

Amemiya (1978) proposed a generalized least squares (AGLS) estimator, which is proved to be consistent and asymptotically efficient. This estimator is available in Stata as a user-written command by Joe Harkness.

```
. search ivtobit, net
Keyword search
        Keywords:  ivtobit
          Search:  (1) Web resources from Stata and from other users
Web resources from Stata and other users
(contacting http://www.stata.com)
1 package found (Stata Journal and STB listed first)
----------------------------------------------------
ivprob-ivtobit from http://fmwww.bc.edu/RePEc/bocode/i
    'IVPROB-IVTOBIT': modules to estimate instrumental variables probit and
    tobit / These programs implement Amemiya Generalized Least Squares (AGLS)
    / estimators for probit and tobit with endogenous regressors.  / Newey
    (J.Metr. 1987, eq. 5.6) provides the formulas used.  The / endogenous
(end of search)
```

We can also perform fit the model manually by two stages: first, regress z on $\mathbf{X}_1$ and $\mathbf{X}_2$, and then fit the tobit model using the predicted values of z from the first stage regression. The estimates of the parameters are proven to be consistent, although the estimated variances are incorrect. We can utilize bootstrapping to obtain proper standard errors. The method is demonstrated as follows:

1. draw bootstrap samples;
2. run first-stage regression $\mathbf{z} = \mathbf{X}_1\delta_1 + \mathbf{X}_2\delta_2 + \mathbf{v}$;
3. calculate the predicted linear combination $\widehat{\mathbf{z}}$;
4. fit the tobit model using $\mathbf{X}_1, \widehat{\mathbf{Z}}$ as explanatory variables;
5. repeat 1–4. Note that the two stage regressions need to be performed on the same bootstrap samples; and
6. compute the standard errors from the sampling distribution of the estimates.

In the simulation, 1,000 random samples are generated for the tobit model, from which the AGLS estimator and bootstrapped standard errors are computed. There are 2 exogenous variables `x11` and `x12`, 1 endogenous variable `z`, which is instrumented by `x21` and `x11`, `x12`. The coefficients in the tobit model are assigned a value of 3 in the simulation. 1,000 bootstrap samples of size 1,000 are drawn. The bootstrap and simulation programs are given below:

```
program mytobit
        version 8
        regress z x11 x12 x21                   // first-stage OLS regression
        predict double zhat, xb                 // prediction
        tobit y x11 x12 zhat, ll(0)             // the tobit regression
end
program myivtobit, rclass                       // to be called by -simulate-
        version 8
        drop _all
        set obs 1000
        generate e1 = invnorm(uniform())
        generate e2 = invnorm(uniform())        // generate error terms
        generate x21 = uniform()
        generate x11 = invnorm(uniform())*1.2
        generate x12 = invnorm(uniform())*1.8
        generate z = 1 + x21 + x11 + x12 + e1   // generate z as endogenous
        generate y = 3 + 3*z + 3*x11 + 3*x12 + e2
        replace y = cond(y>0, y, 0)             // y is censored at 0
        ivtobit y, endog(z) iv(x21) exog(x11 x12) ll(0)
        return scalar x11 = _b[x11]
        return scalar x12 = _b[x12]
        return scalar z = _b[z]
        return scalar cons = _b[_cons]
        return scalar sdx11 = _se[x11]
        return scalar sdx12 = _se[x12]
        return scalar sdz = _se[z]
        return scalar sdcons = _se[_cons]
        bootstrap "mytobit" "_b", reps(1000)
        return scalar bs_x11 = _b[b_x11]
        return scalar bs_x12 = _b[b_x12]
        return scalar bs_z = _b[b_z]
        return scalar bs_cons = _b[b_cons]
        return scalar bs_sdx11 = _se[b_x11]
        return scalar bs_sdx12 = _se[b_x12]
        return scalar bs_sdz = _se[b_z]
        return scalar bs_sdcons = _se[b_cons]
end
```

The results are summarized in Table 3. Wald tests were performed based on the estimated coefficients and standard errors with a significance level of $\alpha = 0.05$. The empirical coverage probabilities are compared with the theoretical level.

Table 3: Monte Carlo simulation results for the tobit model with endogenous variables

A. Estimated Coefficients

		estimated coefficients	
variable	coefficient	AGLS	bootstrap
x1	3	2.999	3.000
x2	3	3.001	3.002
y1	3	3.001	2.998
_cons	3	2.999	3.006

B. Empirical Coverage

		empirical coverage	
variable	$1-\alpha$	AGLS	bootstrap
x1	0.95	0.949	0.963
		(0.933, 0.962)*	(0.949, 0.974)
x2	0.95	0.950	0.961
		(0.944, 0.970)	(0.947, 0.972)
y1	0.95	0.956	0.963
		(0.952, 0.976)	(0.949, 0.974)
_cons	0.95	0.954	0.962
		(0.957, 0.980)	(0.948, 0.973)

*: binomial exact 95% confidence interval

Table 3(A) reports estimated coefficients from the Monte Carlo simulations, which are all quite close to the true value. Table 3(B) shows the rate of not rejecting the null hypothesis: $H_0 : \beta = 3$. The coverage probabilities for the bootstrapped standard errors are slightly higher than for the AGLS standard error for most of the coefficients.

4 Conclusion

This paper discusses the use of the method of bootstrapping as an alternative to obtain standard errors for estimated parameters. The results from Monte Carlo simulations are compared with those from parametric models. Given that the estimated coefficients are consistent, the bootstrap approach reports coverage probabilities as good as parametric methods. Though the examples were chosen for the estimation where other solutions are available in Stata, we can easily extend the application of bootstrapping to other situations. For instance, the third example illustrates a solution for the tobit models with endogenous covariates. We can simply modify the `bootstrap` program for other categorical or limited dependent variable models in the main equation as well. However,

we also need to be cautious when applying the bootstrap method. In the third example, if the first-stage regression is not a linear model, the two-stage estimates will not be consistent, and thus one cannot obtain proper coverage using the bootstrap approach.

In the last two examples, the bootstrapped standard errors tend to be more conservative than the parametric estimates and, hence, give wider coverage for the estimated coefficients. The fact that we only have 1,000 bootstrap samples (500 in the second example) may be a reasonable explanation. The bootstrap sampling distribution approaches the true sampling distribution as the number of resamples gets large. We can reasonably imagine the coverage will be reduced by increasing the number of replications. Unpublished Monte Carlo simulation showed that the coverage probabilities in the second example are dropped to 0.97–0.98 given 100 randomly sampled clusters (of size 5) and 2500 bootstrap replications. However, the coverages will stay at the same level when using more bootstrap repetitions with the sample size unchanged. The empirical coverage probabilities do not approach 0.95 mainly because of the fairly small sample size. While the nonparametric bootstrap method does not rely upon strong assumptions regarding the distribution of the statistic, a key assumption of bootstrapping is the similarity between the characteristics of the sample and of the population. When the sample is of size 500 (100 independent clusters), the assumption of similarity may not be reasonable. The results in Table 2 indicate that the empirical levels tend to converge to 0.95 with increased sample size when the number of bootstrap replications is fixed. In summary, the number of repetitions and sample size both play important roles in the bootstrap method.

5 References

Amemiya, T. 1978. The estimation of a simultaneous equation generalized probit model. *Econometrica* 46(5): 1193–1205.

About the Author

Weihua Guan is a Statistician at Stata Corporation.